




DevOps

▼ Лабораторная работа №1

▼ Условие


 DevOps. Лабораторная работа 1.pdf 45.1KB

▼ Выполнение

- ▶ 1. Установить необходимое ПО
- ▶ 2. Установить apache на VM2
- ▶ 3. Установить nginx на VM1
- ▶ 4. Настроить сетевые интерфейсы на VM1
- ▶ 5. Настроить сетевой интерфейс на VM2
- ▶ 6. Конфигурация nginx на VM1
- ▶ 7. Настроить доступ к VM1 через iptables
- ▶ 8. Настроить доступ к VM2 через iptables
- ▶ 9. Результат

▼ Лабораторная работа №2

▼ Условие

 DevOps. Лабораторная работа 2.pdf 50.2KB

▼ Выполнение

1. Выполнялась лаба с имеющимся проектом с курса по web
2. Для выполнения нужно 2 виртуальные машины
1. Создать репозиторий в GitLab для проекта
2. Загрузить в него проект
3. Если в проекте используется база, то развернуть ее можно на этом сайте:

`customer.elephantsql.com`

`https://customer.elephantsql.com/instance`

4. Первая виртуальная машина

a. Подключение через ssh, поэтому в базисе нужно в port forwarding добавить новый порт с параметрами: внешние — одинаковые и какие захотите, а внутренний — 22

b. Подключаемся по ssh:

```
ssh user@ 176.118.165.63 -p <ваш внешний порт>
```

c. Установить gitlab-runner

```
curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh" | sudo bash  
sudo apt-get install gitlab-runner
```

d. Зарегистрировать gitlab-runner

i. Команда `sudo gitlab-runner register`

ii. В url: <https://git.iu7.bmstu.ru/>

iii. Токен: в репозитории → settings → CI/CD → runners (expand). Там лежит ваш токен

iv. Описание: любое

v. Теги: build,test,deploy

vi. Следующий шаг пропускаем

vii. Исполнитель: shell

viii. Готово (можно в репе проверить, что он есть)

e. Установить dotnet (для бекенда C# приложений): `sudo apt-get update && sudo apt-get install -y dotnet-sdk-6.0`

f. Установить npm (для фронтенда): `sudo apt install npm`

g. Установить sshpass: `sudo apt install sshpass`

- h. В корне проекта создать файл `.gitlab-ci.yml` и заполнить его. Мой файл представлен ниже (нужно добавить еще в стадию деплоя прокидывание конфига `nginx.conf`, у меня это не сделано). При этом добавить в веб версии нужные переменные.

```
stages: - build - test - deploy build-backend: stage: build
tags: - build script: - cd backend/ - dotnet publish -f net6.0
-c Release -o ./build artifacts: untracked: false when:
on_success expire_in: "1 hour" paths: - backend/build build-
frontend: stage: build tags: - build script: - cd frontend/ -
npm i - npm run build - mv dist build artifacts: untracked:
false when: on_success expire_in: "1 hour" paths: -
frontend/build test-backend-unit-bl: stage: test tags: - test
script: - cd test/UnitBL/ - dotnet test test-backend-unit-db:
stage: test tags: - test script: - cd test/UnitDB/ - dotnet
test deploy: stage: deploy tags: - test script: - sshpass -p
$PROD_PASSWORD rsync -e "ssh -o StrictHostKeyChecking=no" -a
backend/build/ $PROD_USER@$PROD_HOST:/var/opt/app - sshpass -p
$PROD_PASSWORD rsync -e "ssh -o StrictHostKeyChecking=no" -a -
-delete frontend/build/ $PROD_USER@$PROD_HOST:/var/www/site -
echo $PROD_PASSWORD | sshpass -p $PROD_PASSWORD ssh
$PROD_USER@$PROD_HOST sudo -S systemctl restart servering-
backend.service rules: - if: '$CI_COMMIT_REF_NAME == "dev" ||
$CI_COMMIT_REF_NAME == "main"' when: manual - when: never
```

- i. Готово

5. Вторая виртуальная машина

a. Подключение через ssh, поэтому в базисе нужно в port forwarding добавить новый порт с параметрами: внешние — одинаковые и какие захотите, а внутренний — 22

b. Подключаемся по ssh:

```
ssh user@ 176.118.165.63 -p <ваш внешний порт>
```

c. Установить dotnet (для бекенда C# приложений): `sudo apt-get update && sudo apt-get install -y dotnet-sdk-6.0`

d. Создать папку для фронтенда: `sudo mkdir /var/www/site` и выдать на нее права пользователю: `sudo chown -R $USER /var/www/site`

e. Создать папку для бекенда: `sudo mkdir /var/opt/app` и выдать на нее права пользователю: `sudo chown -R $USER /var/opt/app`

▼ Мои сохраненные данные

SSH для виртуалок (моих):

1. Установить nginx: `sudo apt install nginx`

2. Ip: 176.118.165.63:8090 nginx.conf. Мой конфиг:

Name: user

Password: 8XL2Yu5eT

3. Ip: 176.118.165.63:9191

worker_processes 1; pid /var/run/nginx.pid; events {

worker_connections 1024; } http { include

Name: user

/etc/nginx/mime.types; sendfile on; keepalive_timeout 65;

Password: 8XL2Yu5eT

server { listen 80; server_name localhost; root /var/www/site;

▼ Лабораторная работа №3